

PAT-NO: JP411282664A

DOCUMENT-IDENTIFIER: JP 11282664 A

TITLE: **INCOMPATIBLE** PROGRAM CORRECTION SYSTEM

PUBN-DATE: October 15, 1999

INVENTOR-INFORMATION:

NAME	COUNTRY
TAGO, SHIGERU	N/A

ASSIGNEE-INFORMATION:

NAME	COUNTRY
HITACHI SOFTWARE ENG CO LTD	N/A

APPL-NO: JP10081098

APPL-DATE: March 27, 1998

INT-CL (IPC): G06F009/06

ABSTRACT:

PROBLEM TO BE SOLVED: To attain a desired function by confirming automatically or manually a submodule that is **incompatible** and replacing it with a *prepared compatible module*.

SOLUTION: A transfer request of a check processing module is sent to a server 111 via a network 104. An arithmetic unit 102 reads the relevant submodule out of an auxiliary storage device 103 and executes the received check processing module. If the submodule stored in the storage device 103 is **incompatible**, the identifier of a compatible module is designated for the **incompatible** module. Then a transfer request of the compatible module is given to the server 111. The server 111 refers to a table to take a module file of the designated identifier out of an auxiliary storage device 112 based on the module file name for the submodule that is designated by the transfer request and then transfers the module file.

COPYRIGHT: (C)1999,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-282664

(43) 公開日 平成11年(1999)10月15日

(51) Int.Cl.⁶
G 0 6 F 9/06

識別記号
5 4 0

F I
G 0 6 F 9/06

5 4 0 G

審査請求 未請求 請求項の数7 O L (全 8 頁)

(21) 出願番号 特願平10-81098

(22) 出願日 平成10年(1998) 3月27日

(71) 出願人 000233055

日立ソフトウェアエンジニアリング株式会
社

神奈川県横浜市中区尾上町 6 丁目 81 番地

(72) 発明者 多胡 滋

神奈川県横浜市中区尾上町 6 丁目 81 番地
日立ソフトウェアエンジニアリング株式会
社内

(74) 代理人 弁理士 秋田 収喜

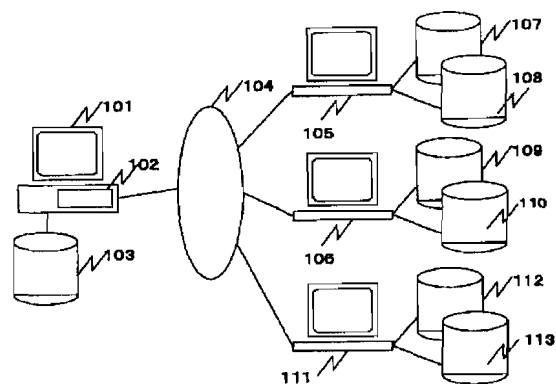
(54) 【発明の名称】 非互換プログラム修正システム

(57) 【要約】

【課題】 計算機システムの実行環境に記憶されているサブ・モジュールに非互換性があった場合にも、目的の機能を実現可能にすること。

【解決手段】 あるプログラム・コードをある計算機システム上で実行するのに先立って、この計算機システムが記憶しているサブ・モジュールのうち、プログラム・コードが前提としている機能あるいは副作用と異なる非互換モジュールをリストアップする検査機能と、前記機能によりリストアップされる可能性のあるサブ・モジュールについて、同じパラメータを入力とする別の互換モジュールを登録および検索する機能とを持たせ、前記検査機能により、リストアップされたサブ・モジュール群の各々について、それと同じ名前で参照され、同じパラメータを入力とする、前記データベース機能により登録されたサブ・モジュールによって置換した上で、前記プログラム・コードを実行する。

図 1



【特許請求の範囲】

【請求項1】 演算装置上で任意のプログラム・コードを実行可能な計算機システムであり、かつ前記プログラム・コードが、1つあるいは複数のサブ・モジュールと、1つのメイン・モジュールとから構成され、かつ複数の計算機システムの各々が、同じ名前で参照され、同じパラメータを入力とするサブ・モジュールから構成される、前記プログラム・コードの実行に必要なサブ・モジュール群の一部あるいは全部を予め記憶し、前記プログラム・コードの実行時には、前記メイン・モジュールのみを導入した上で、既に記憶済みのサブ・モジュールを組み合わせることによってプログラム・コード全体を構成し、実行する機能を持つ計算機システムにおいて、前記プログラム・コードの開発時に前提となっている機能あるいは副作用を持つサブ・モジュールに対して、予め各計算機システムが記憶している同じ名前で参照され、同じパラメータを入力とするサブ・モジュールで、その機能あるいは副作用が完全には同一でない場合に、該当するサブ・モジュールを、前記プログラム・コードの実行に先立ってリストアップする手段と、該当するサブ・モジュールと同じ名前で参照され、同じパラメータを入力とし、前記プログラム・コードの開発時に前提となっている機能あるいは副作用を持つサブ・モジュールを登録および検索する手段とを備え、機能あるいは副作用が完全には同一でないサブ・モジュールに対して、機能あるいは副作用が完全に同一であるサブ・モジュールを検索し、これを前記サブ・モジュールと置換した上で前記プログラム・コードを実行することにより、同じ名前で参照され、同じパラメータを入力とし、機能あるいは副作用が完全には同一ではないサブ・モジュールを記憶している計算機システム上で、前記プログラム・コードを正常に実行させることを特徴とするプログラム修正システム。

【請求項2】 実行するプログラム・コードの前提となっている機能あるいは副作用を持つサブ・モジュールに対して、各計算機が記憶している同じ名前で参照され、同じパラメータを入力とするサブ・モジュールで、その機能あるいは副作用が完全には同一でないようなサブ・モジュールを、前記プログラム・コードの実行に先立って自動的にリストアップする手段を備えることを特徴とする請求項1記載のプログラム修正システム。

【請求項3】 実行するプログラム・コードの前提となっている機能あるいは副作用を持つサブ・モジュールに対して、各計算機が記憶している同じ名前で参照され、同じパラメータを入力とするサブ・モジュールで、その機能あるいは副作用が完全には同一でないようなサブ・モジュールを、前記プログラム・コードの実行に先立って該当する機能あるいは副作用を実行し、その結果を実行者に出力装置を介して確認させることによってリストアップする手段を備えることを特徴とする請求項1記載

のプログラム修正システム。

【請求項4】 ネットワーク経由での置換任意のサブ・モジュールを登録および検索する機能と、ネットワークを介して任意のサブ・モジュールを転送する機能を持つサーバシステムと、ネットワークを介して受信したサブ・モジュールとメイン・モジュールを組み合わせてプログラム・コードを実行する機能を持つクライアントシステムを備え、クライアントシステムが記憶しているサブ・モジュールと同じ名前で参照され、同じパラメータを入力とするサブ・モジュールをサーバシステムから転送することを特徴とする請求項1記載のプログラム修正システム。

【請求項5】 互換性確認モジュールの添付サーバシステムに登録済みの特定のサブ・モジュールに対し、同じ名前で参照され、同じパラメータを入力とする別のサブ・モジュールが、同一の機能あるいは副作用を持つことを確認するためのプログラム・コードを前記サブ・モジュールと対応させてサーバシステム上に登録し、任意のプログラム・コードの実行に先立ってサブ・モジュールの機能あるいは副作用を確認する際に、前記サーバシステム上のプログラム・コードを実行し、その結果を元にサブ・モジュールのリストアップをすることを特徴とする請求項2～4記載のいずれかのプログラム修正システム。

【請求項6】 任意のプログラム・コードに対して、このプログラム・コードの実行に必要な全サブ・モジュールのうち、実行に先立って機能あるいは副作用の確認対象とするサブ・モジュール群を、前記プログラム・コードと1対1に対応付けて指定することを特徴とする請求項2～4記載のいずれかのプログラム修正システム。

【請求項7】 同じ名前で参照され、同じパラメータを入力とするサブ・モジュール群の各々について、これと置換可能な複数のサブ・モジュールに対し、その各々に異なる機能あるいは副作用を持たせ、任意のプログラム・コードの実行に先立って、いずれかのサブ・モジュールの置換操作が必要になった時点で、予めプログラム・コードごとに指定されている機能あるいは副作用を持つサブ・モジュールを選択し、置換することによって、前記プログラム・モジュールが開発時に前提としている機能あるいは副作用のみを正常に動作させることを特徴とする請求項2～4記載のいずれかのプログラム修正システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、演算装置上で任意のプログラム・コードを実行可能な計算機システムに関わり、特に、前記プログラム・コードが、1つあるいは複数のサブ・モジュールと、1つのメイン・モジュールとから構成され、かつ複数の計算機システムの各々が、同じ名前で参照され、同じパラメータを入力とするサブ

・モジュールから構成される、前記プログラム・コードの実行に必要なサブ・モジュール群の一部あるいは全部を予め記憶し、前記プログラム・コードの実行時には、前記メイン・モジュールのみを導入した上で、すでに記憶済みのサブ・モジュールを組み合わせることによってプログラム・コード全体を構成し、実行する機能を持つ計算機システムにおけるプログラム修正システムに関するものである。

【0002】

【従来の技術】演算装置上で任意のプログラム・コードを実行可能な計算機システムにおいて、あるプログラム・コードを実行しようとする場合、そのプログラム・コードが利用する全てのモジュール（メイン・モジュールを含む）を1つのプログラム・パッケージとして配布あるいは複写し、それを導入するという方法では、異なるプログラム・コード間で同一のサブ・モジュール（通常、ライブラリと呼ばれる）を利用する機会が多いことを考えると、異なるプログラム・パッケージのファイル内に共通のサブ・モジュール・ファイルが含まれることになり、プログラム・パッケージの配布あるいは複写の際に無駄な負荷がかかったり、導入先の補助記憶装置の記憶域の大きさが無駄に大きくなったりするなどの問題がある。

【0003】そこで、多くのプログラム・コード間で共通に利用される可能性の高いサブ・モジュールについては、1つだけを予め前記計算機システム上に記憶しておき、前記プログラム・コードの実行時に、プログラム・コードごとに記憶されているメイン・モジュールおよびサブ・モジュールと、予め記憶されているサブ・モジュールの中から実行に必要なものを取り出して組み合わせることにより、プログラム・コード全体を動的に構成して実行することにより、無駄な負荷や補助記憶装置の記憶域の大きさの無駄を省略する方法が、多くの計算機システムで採用されている。これを通常、ダイナミック・リンク・ライブラリと呼ぶ。

【0004】

【発明が解決しようとする課題】しかし、前記方法を採用した計算機システムにおいては、次のような問題がある。

【0005】すなわち、異なる計算機システム上に予め記憶されているサブ・モジュールのうち、同じ名前参照され、同じパラメータを入力とするサブ・モジュールについて、その機能あるいは副作用が、各計算機システム間で異なるという問題が発生する。副作用とは、サブ・モジュールが動作することにより、他のモジュールに及ぼす影響などの主たる作用に付随して起こる作用のことである。これを一般に非互換性と呼ぶ。これは、同じ名前参照され、同じパラメータを入力とするサブ・モジュールであっても、その開発時期が異なっていたり、共通でない不具合（バグ）を含んでいたたり、各サブ・モ

ジュールが利用する別のサブ・モジュールやオペレーティング・システムの機能や副作用に違いがあるなどの理由に起因する。

【0006】そのため、あるプログラム・コードの開発者が、そのプログラム・コードの実行時に利用するサブ・モジュールの機能あるいは副作用として開発時点で前提にしているものと、実際に実行される計算機システムが記憶しているサブ・モジュールの機能あるいは副作用との間に相違点があった場合に、前記プログラム・コードの開発者が目的としている動作を実現できない計算機システムが存在してしまうという問題がある。

【0007】本発明の目的は、このような問題を解決することであり、プログラムパッケージに含まれていないサブ・モジュールに非互換性がある場合に、それを自動的あるいは手動操作で確認し、予め用意されている互換モジュールと置換することにより、実行環境に記憶されているサブ・モジュールに非互換性があった場合にも、目的の機能を実現することができる非互換プログラム修正システムを提供することにある。

【0008】

【課題を解決するための手段】本発明では、以下のような手段により、上記の課題を解決する。

【0009】すなわち、あるプログラム・コードをある計算機システム上で実行するのに先立って、この計算機システムが記憶しているサブ・モジュールのうち、その機能あるいは副作用が前記プログラム・コードが前提としている機能あるいは副作用と異なるもの（非互換モジュール）をリストアップする検査手段と、リストアップされる可能性のあるサブ・モジュールについて、前記プログラム・コードが前提としている機能あるいは副作用を持ち、同じ名前参照され、同じパラメータを入力とする別のサブ・モジュール（互換モジュール）を登録および検索する手段とを、前記計算機システムあるいは、前記計算機システムとネットワークを介して通信可能なサーバシステムに持たせ、前記検査手段により、前提としている機能あるいは副作用と異なる機能あるいは副作用を持つものとしてリストアップされたサブ・モジュール群の各々について、それと同じ名前参照され、同じパラメータを入力とする互換モジュールによって置換した上で、前記プログラム・コードを実行する。

【0010】さらに、前記検査手段においては、各サブ・モジュールごとの予想される相違点に従って予め用意された検査機構により、前記プログラム・コードの実行直前に自動的に検査処理を実行する機能と、各サブ・モジュールごとの予想される相違点に従って予め用意されたパラメータを該当するサブ・モジュールに入力し、その結果を出力装置に出力した上で、前提となっている結果が得られたかどうかを実行者によって確認させ、確認結果を折り返し入力させることにより検査処理を実行する機能の、いずれかあるいはその両方を持たせる。

【0011】さらに、前記2種類の検査機能の各々について、個々のサブ・モジュールに対応した検査処理を実行するモジュールを、データベース機能により、各サブ・モジュールに対応付けて登録および検索することにより、検査処理実行時点で検査対象となるサブ・モジュールに対応した検査処理モジュールを取り出し、起動する機能を持たせる。

【0012】さらに、検査機能により、前提としている機能あるいは副作用と異なるものとしてリストアップされる可能性のあるサブ・モジュールのうち、あるプログラム・コードにとって、置換する必要があるものかどうかを指定する機能を、前記プログラム・コードに持たせる。

【0013】さらに、前記データベース機能により登録されるサブ・モジュールについて、同じ名前で参照され、同じパラメータを入力とするサブ・モジュールに対し、異なる機能あるいは副作用を持つ複数のサブ・モジュールを登録することにより、あるプログラム・コードの実行に当たって、そのプログラム・コードが必要としている機能あるいは副作用を持つサブ・モジュールを、データベース機能により取り出し、置換する機能を持たせる。

【0014】本発明の手段により、以下のことが可能となる。

【0015】1) あるプログラム・コードを実行しようとした際、実行する計算機システム上に予め記憶されているサブ・モジュールの中に、前記プログラム・コードが前提としている機能あるいは副作用を持たないサブ・モジュール(非互換モジュール)があった場合に、これと、互換モジュールとを自動的にあるいは手動操作により置換し、その上で前記プログラム・コードを実行することにより、実行する計算機システムによって非互換モジュールが存在したとしても、目的の動作を実現することが可能となる。

【0016】2) また、非互換モジュールの存在が新たに確認されたとしても、データベースに対し、そのサブ・モジュールの検査処理を実行するモジュールと、そのサブ・モジュールと置換する互換モジュールを登録することにより、未確認の非互換モジュールの記憶されている計算機システム上でも目的の動作を実現することが可能となる。

【0017】3) また、全ての非互換モジュールの中から、置換する必要があるサブ・モジュールを指定することにより、非互換モジュールの置換に必要な処理の負荷や補助記憶装置の記憶域を最適化することができる。

【0018】4) また、同一の非互換モジュールについて、複数のプログラム・コードの各々が必要としている機能あるいは副作用についてのみ互換性を実現した複数のサブ・モジュールを利用可能にすることで、置換するサブ・モジュールの大きさを小さくし、その置換に必要

な処理の負荷や、そのサブ・モジュールを開発するのに必要な工数などを削減することが可能となる。

【0019】

【発明の実施の形態】以下、本発明の実施の形態を図面を参照して詳細に説明する。

【0020】図1は、本発明の実施の形態を示すシステム構成図である。この実施形態は、本発明をクライアント・サーバ構成の計算機システムに適用したものである。

【0021】図1において、101は、文字あるいは画像を表示する機能を持つディスプレイ装置、102は、プログラム・コードを実行する機能を持つ演算装置である。103は、本計算機システム上で実行されるプログラム・コードに必要な各種サブ・モジュールを記憶している補助記憶装置である。

【0022】104は、2個の計算機システム間でのデータ通信を可能とするネットワークである。105および106は、ネットワーク104を介してデータの送受信を行う機能を持つサーバ・システムである。107および109は、演算装置102上で動作可能なプログラム・コードを記憶するための補助記憶装置である。

【0023】108および110は、非互換モジュールであるかどうかの検査の対象とするサブ・モジュールのリストを記憶するための補助記憶装置である。

【0024】111は、ネットワーク104を介してデータの送受信を行う機能を持つサーバ・システムである。112は、各種互換モジュールを記憶するための補助記憶装置である。113は、各サブ・モジュールが非互換モジュールであるかどうかを検査する処理を実行するための各種検査処理モジュールを記憶するための補助記憶装置である。

【0025】図2は、計算機システム上で動作するプログラム・コードのメイン・モジュールが利用するサブ・モジュールの名前と、それらに対する入力パラメータのテーブルを示している。このテーブルは、前記プログラム・コード中に含まれている。

【0026】201は、サブ・モジュールの名前を示している。202～204は、201に記述した名前のサブ・モジュールに対し入力するパラメータである。

【0027】図3は、補助記憶装置103から読み出した各種サブ・モジュールのメモリ中の一時保管場所を示すためのテーブルである。

【0028】図3において、301は、サブ・モジュールの名前を示している。302は、メモリ中の対応するサブ・モジュールの記憶場所、すなわちアドレスを示している。この記憶場所は、メモリのハードウェア・アドレスでなくとも、高級言語による変数あるいはポインタなども可能である。

【0029】図4は、特定のプログラム・コードに対し、その実行に先立って、非互換モジュールであるかど

うかを検査すべきサブ・モジュール群を示すテーブルである。図4において、401は、サブ・モジュールの名前を示している。402～404は、401に記述した名前のサブ・モジュールに対し、検査時に入力するパラメータである。405は、204に記述した名前のサブ・モジュールの検査処理を実行させるための検査処理モジュールの識別子(ID)である。

【0030】図5は、補助記憶装置113に記憶されている各種検査処理モジュールの一覧を格納したテーブルである。図5において、501は、サブ・モジュールの名前を示している。502は、501に記述したサブ・モジュールの複数ある互換モジュールを識別するための識別子(ID)である。503は、501に記述したサブ・モジュールのモジュール・ファイル(プログラムファイル)のファイル名を示している。

【0031】図6は、補助記憶装置113に記憶されている各種検査処理モジュールの1つが、対応するサブ・モジュールの検査を行う際にディスプレイ装置101に表示する画面の例である。

【0032】図7は、図4のテーブルに従い検査した結果、非互換モジュールと判断されたサブ・モジュール群を示すテーブルである。図7において、701は、サブ・モジュールの名前を示している。702は、非互換モジュールと判断された701に記述したサブ・モジュールと置換する補助記憶装置112に記憶されているサブ・モジュールの識別子(ID)を示している。

【0033】図8は、補助記憶装置112に記憶されているサブ・モジュールの一覧を格納したテーブルである。801は、サブ・モジュールの名前を示している。802は、801に記述したサブ・モジュールの複数ある互換モジュールを識別するための識別子(ID)である。803は、各互換モジュールのモジュール・ファイル(プログラムファイル)のファイル名を示している。

【0034】図9は、本実施形態における処理の流れを示すフローチャートである。以下、このフローチャートに従い、本実施形態の全体の処理を説明する。

【0035】まず、演算装置102において、特定のプログラム・コードを指定して、計算機システムを起動する(ステップ901)。すると、指定されたプログラム・コードを補助記憶装置に記憶しているサーバ、例えば106に対してネットワーク104経由で、プログラム・コードの転送要求が送信される。そして、補助記憶装置110からプログラム・コードが読み出され、やはりネットワーク104経由で演算装置103上にプログラム・コードが複写される(ステップ902)。このプログラム・コードには図2のテーブルも含まれている。次に、同じサーバ106から、図4に示したテーブルを、手順902と同様の方法で、演算装置103上に複写する(ステップ903)。これにより、指定されたプログラム・コードに対応するメイン・モジュールが使用する

サブ・モジュールのリスト(図2)と、検査対象となるサブ・モジュールのリスト(図4)が得られたことになる。以下、図4に示したテーブルに一覧されているサブ・モジュールの各々に対して、一連の処理を実行する。

【0036】まず、図4の401からサブ・モジュール名を読み出し、また、その入力パラメータを402～404から読み出す(ステップ904)。次に、そのサブ・モジュールに対して置換する互換モジュールの識別子を405から読み出す(ステップ905)。次に、ネットワーク104経由で、サーバ111に対し、検査処理モジュールの転送要求を送信する(ステップ906)。サーバ111は、図5に示したテーブルを参照し、モジュール名501と識別子502が転送要求で指定されたものと一致するサブ・モジュールについて、503に記述されたモジュール・ファイルを補助記憶装置113から読み出し、転送する(ステップ907)。

【0037】演算装置102では、補助記憶装置103から、該当するサブ・モジュールを読み出し、ステップ907で受信した検査処理モジュールを実行する(ステップ908)。検査処理モジュールの実行に当たっては、そこからさらに起動される補助記憶装置103に記憶されていたサブ・モジュールに対し、402～404に記述されている入力パラメータを入力する。

【0038】検査処理モジュールは、908の入力パラメータを、対応するサブ・モジュールに入力し、それを起動する。そして、その結果(機能あるいは副作用)を検査する(ステップ909)。この検査においては、サブ・モジュールの終了時に出力される文字データや数値データを期待値と比較したり、図6に示したように、例えばラベルの文字列が正しく表示されるかどうかを実行者に問い合わせ、その結果を「OK」なのか「NOTOK」なのかで異なるボタンを押させることにより実行する。図6においては、「ラベルの文字列の表示」というのが正しいのであるが、破線枠に示す「示」の文字が欠落している例を示している。この例では、「示」の文字が欠落しているため、実行者は「NOTOK」のボタンを選択する。

【0039】その結果、期待値を異なる出力を得たり、実行者が「NOTOK」のボタンを押したりした場合は、補助記憶装置103に記憶されているサブ・モジュールが非互換モジュールであると判断し、図7に示したテーブルに、サブ・モジュールの名前と、互換モジュールの識別子405を追加する(ステップ910)。

【0040】もし、非互換モジュールでないと判断された場合は、補助記憶装置103に記憶されているサブ・モジュールを演算装置102のメモリ中に読み込み、その記憶場所を302に追加する(ステップ911)。

【0041】こうして、ステップ904～911の処理を、図4に示したテーブルに一覧されているサブ・モジュールの全てについて実行する。この結果、作成された

図7のテーブルに一覧されているサブ・モジュールの各々について、サブ・モジュールの名前701と、そのサブ・モジュールの互換モジュールの識別子702を指定して、サーバ111に対し、互換モジュールの転送要求を送信する(ステップ912)。

【0042】サーバ111は、図8のテーブルを参照し、912の転送要求で指定されたサブ・モジュールの、指定された識別子のモジュール・ファイルをモジュールファイル名803を元に補助記憶装置112から取り出し、転送する(ステップ913)。転送されたモジュール・ファイルは、メモリ中に読み込まれ、その記憶場所を302に追加する(ステップ914)。最後に、図2に示されていて図4に示されていないサブ・モジュールを補助記憶装置103から読み出し、その記憶場所を302に追加する(ステップ915)。そして、演算装置102上で、図2を参照しながら、メイン・モジュールおよび各サブ・モジュールを実行する(ステップ916)。

【0043】なお、本実施形態は、クライアント・サーバ構成の計算機システムに本発明を適用したものであるが、本発明はこのような構成に限定されるものではなく、上記実施形態と同様の機能を1つの計算機システム内に設けて実施することができる。また、サーバ機能を1つのサーバに集約した構成にすることもできる。このほかに、実施に当たっては、メモリ容量や負荷を考慮し、各機能を適宜に分散して構成することができる。

【0044】

【発明の効果】以上のように、本発明によれば、次のような効果が得られる。

1) あるプログラム・コードを実行しようとした際、実行する計算機システム上に予め記憶されているサブ・モジュールの中に、前記プログラム・コードが前提としている機能あるいは副作用を持たないサブ・モジュール(非互換モジュール)があった場合に、これと、互換モジュールとを自動的あるいは手動操作により置換し、その上で前記プログラム・コードを実行することにより、実行する計算機システムによって非互換モジュールが存在したとしても、目的の動作を実現することが可能となる。

【0045】2) また、非互換モジュールの存在が新たに確認されたとしても、データベースに対し、そのサブ

・モジュールの検査処理を実行するモジュールと、そのサブ・モジュールと置換する互換モジュールを登録することにより、未確認の非互換モジュールの記憶されている計算機システム上でも目的の動作を実現することが可能となる。

【0046】3) また、全ての非互換モジュールの中から、置換する必要のあるサブ・モジュールを指定することにより、非互換モジュールの置換に必要な処理の負荷や補助記憶装置の記憶域を最適化することができる。

【0047】4) また、同一の非互換モジュールについて、複数のプログラム・コードの各々が必要としている機能あるいは副作用についてのみ互換性を実現した複数のサブ・モジュールを利用可能にすることで、置換するサブ・モジュールの大きさを小さくし、その置換に必要な処理の負荷や、そのサブ・モジュールを開発するのに必要な工数などを削減することが可能となる。

【図面の簡単な説明】

【図1】本発明の一実施形態を示すシステム構成図である。

【図2】メイン・モジュールから利用されるサブ・モジュールとその入力パラメータのテーブル構成図である。

【図3】メモリ中の各サブ・モジュールの名前とその記憶場所のテーブル構成図である。

【図4】特定のプログラム・コードに関して、検査する必要があるサブ・モジュールとその入力パラメータのテーブル構成図である。

【図5】各サブ・モジュールに対応した検査処理モジュールの名前と識別子のテーブル構成図である。

【図6】検査処理の出力例を示す図である。

【図7】置換が必要なサブ・モジュールの名前と識別子のテーブル構成図である。

【図8】置換するための互換モジュールの名前と識別子のテーブル構成図である。

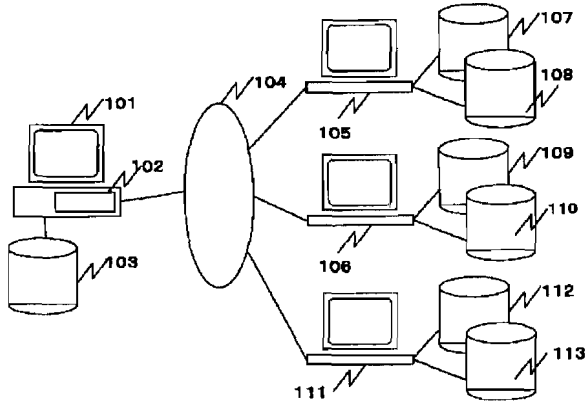
【図9】実施形態の処理を示すフローチャートである。

【符号の説明】

101…ディスプレイ装置、102…演算装置、103…補助記憶装置、104…ネットワーク、105、106サーバシステム、107～110…補助記憶装置、111…サーバシステム、112、113…補助記憶装置。

【図1】

図1



【図2】

図2

モジュール名	パラメータ1	パラメータ2	パラメータ3
Canvas			
Frame	"フレーム"		
Socket	"サーバ"	ポート番号	

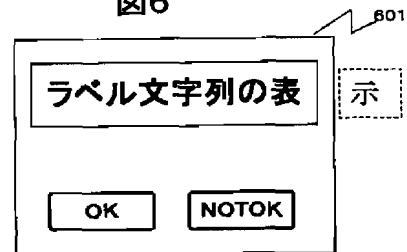
【図3】

図3

モジュール名	記憶場所
Canvas	1A2B3C
Frame	3C2B1A
Socket	9F8B7D

【図6】

図6



【図4】

図4

モジュール名	パラメータ1	パラメータ2	パラメータ3	識別子
Canvas				10
Frame	"フレーム"			20
Socket	"サーバ"	ポート番号		30

【図5】

図5

モジュール名	識別子	モジュール・ファイル
Canvas	10	CanvasT1. class
	11	CanvasT2. class
	12	CanvasT3. class
Frame	20	FrameT1. class
	21	FrameT2. class
	22	FrameT3. class
Socket	30	SocketT1. class
	31	SocketT2. class
	32	SocketT3. class

【図7】

図7

モジュール名	識別子
Canvas	10
Frame	20
Socket	30

【図8】

図8

モジュール名	識別子	モジュール・ファイル
Canvas	10	Canvas1. class
	11	Canvas2. class
	12	Canvas3. class
Frame	20	Frame1. class
	21	Frame2. class
	22	Frame3. class
Socket	30	Socket1. class
	31	Socket2. class
	32	Socket3. class

【図9】

図9

